



Configurateur de Système Automatisé de Production en vue de la Formation de ses Utilisateurs

Saher Arnous, Arnaud Lelevé, Patrick Prévot, Khalid Kouiss

► To cite this version:

Saher Arnous, Arnaud Lelevé, Patrick Prévot, Khalid Kouiss. Configurateur de Système Automatisé de Production en vue de la Formation de ses Utilisateurs. 12e colloque national AIP-PRIMECA, Mar 2011, Le Mont Dore, France. pp.6. hal-00581627

HAL Id: hal-00581627

<https://hal.science/hal-00581627>

Submitted on 1 Jun 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONFIGURATEUR DE SYSTEME AUTOMATISE DE PRODUCTION EN VUE DE LA FORMATION DE SES UTILISATEURS

Saher ARNOUS (1), Arnaud LELEVE (2), Patrick PREVOT (1), Khalid KOUISS (3)

- (1) LIESP, INSA, Université de Lyon, F69621, Villeurbanne cedex, saher073@gmail.com
(2) AMPERE, INSA, Université de Lyon, F69621, Villeurbanne cedex, arnaud.leleve@insa-lyon.fr
(3) LIMOS, IFMA, Campus Scientifique des Cézeaux BP 265, F63170 Aubière, kouiss@ifma.fr

Résumé :

Cet article présente l'avancement d'une recherche visant à proposer un logiciel de configuration d'un Système Automatisé de Production en fonction d'objectifs pédagogiques. Ce type de logiciel trouve son utilité à la fois dans un cadre industriel quand il s'agit de former le personnel à l'usage d'un nouveau SAP, et dans un cadre universitaire lorsque des SAP complexes sont partagés entre différentes formations comme c'est le cas dans de nombreux AIP. Le contexte scientifique de ce travail est présenté et les besoins fonctionnels de ce système affinés. L'architecture globale est décrite et les choix de réalisation ainsi que la procédure d'expérimentation (à venir) sont exposés.

Mots clés : Conception collaborative des produits et des systèmes, Méthodologie de conception des systèmes, Modélisation de connaissances, Prise en compte des aspects humains, conception orientée Composants, E-Learning

1 Introduction

Lors de l'installation d'un nouveau Système Automatisé dans un atelier de production, de nombreuses heures sont consacrées à la formation du personnel de production et de maintenance pour qu'ils le prennent en main. Cette formation a lieu souvent sur le terrain, d'abord hors ligne puis au fur et à mesure, en production. Cet apprentissage est à la fois théorique (procédures de fonctionnement, de remise en service, de maintenance, ...) et pratique (sur la machine elle-même avec des produits factices puis réels). La formation pratique consiste à reproduire des cas d'utilisation typiques que les utilisateurs rencontreront ultérieurement. Pour ce faire, il est nécessaire de « placer » le système dans des conditions d'usage faisant apparaître des phénomènes (pannes, blocages, ...) permettant ensuite de travailler sur les réactions appropriées des personnels. Si cette reconfiguration du système pouvait être « automatisée » afin de reproduire systématiquement ce type de situations à but pédagogique, le formateur pourrait enchaîner des scénarios d'usage en se concentrant sur cet apprentissage et en limitant ces temps de reconfiguration. D'autre part, cela permettrait également aux industriels de reproduire ces cas d'utilisation ultérieurement, en l'absence du formateur initial, pour rafraîchir la mémoire de leurs personnels et les maintenir à un niveau de compétences toujours à jour.

D'autre part, un fabricant de machines fournit des machines généralement très proches les unes des autres, composées de sous-systèmes réutilisés en fonction des besoins spécifiques de chaque client. Le fabricant monte donc régulièrement des formations assez proches les unes des autres, les différences résidant dans la présence et la version des sous-systèmes composant ses machines. Ainsi, si l'outil de

reconfiguration présenté précédemment pouvait se baser sur la nomenclature de la machine sur laquelle il faut créer une nouvelle formation, le fabricant gagnerait en temps de scénarisation car le reconfigurateur pourrait déjà proposer une ossature de formation assemblée à partir des sous-ensembles de la machine, ossature cohérente avec les composants et leurs versions respectives.

Dans un autre contexte, académique, les AIP-Primeca¹ hébergent de nombreux systèmes industriels utilisés par les centres universitaires locaux afin de former leurs élèves à leur conception, leur réalisation, leur réparation, ou simplement leur utilisation selon différents objectifs pédagogiques (MES, ordonnancement, ...). Afin de compenser les lourds coûts d'acquisition et de maintenance de ces systèmes, les AIP cherchent à les ouvrir, comme application pratique, à un maximum de disciplines. Un usage massif de tels systèmes pour des besoins différents engendre cependant des temps de reconfiguration qui peuvent être problématiques, surtout lorsque nombre de leurs utilisateurs ne sont pas automaticiens. En faisant ainsi appel à cet outil de reconfiguration, les enseignants pourraient être accompagnés, au moment de la scénarisation, d'automaticiens experts sur chaque système, afin de créer une banque de scénarios répondant à des besoins pédagogiques prédéfinis. Ils auraient ensuite la possibilité de commuter d'un scénario à un autre sans nécessiter la présence systématique d'automaticiens. Ainsi le même outil de reconfiguration augmenterait la flexibilité (nécessaire à un usage intensif) de ces installations, et donc leur attractivité pour de nouvelles applications et de nouveaux publics (démonstrations, TP à distance, ...).

Ce travail vise donc à concevoir un tel logiciel de reconfiguration, à en développer un prototype et à en caractériser les apports effectifs. Cet article aborde donc les bases scientifiques de ce projet, en précise les besoins fonctionnels et en présente l'architecture globale résultante. Les principaux choix de réalisation d'un prototype et la programmation de son expérimentation sont ensuite décrits.

2 Environnement scientifique de ce projet

Ce projet se situe à cheval entre la conception de systèmes automatisés et le *E-Learning*.

Fondements liés au *E-Learning*

Que ce soit dans un cadre industriel ou académique, ce système se caractérise par un objectif pédagogique indirect : aider le formateur à préparer son outil de travail, qui sera au centre de ses objectifs pédagogiques. Ce système rentre donc dans le cadre pédagogique des *Travaux Pratiques (TP)*. Nous cherchons, à travers ce projet à transformer et rationaliser des processus de reconfiguration qui existent déjà mais qui sont la plupart du temps manuels (donc lents), empiriques et sous - ou pas du tout - documentés. En effet, dans un cadre académique, l'automaticien responsable du système est malheureusement trop souvent le seul à centraliser l'expertise liée à ce système, ce qui induit un risque très fort d'immobilisation du système en cas d'absence et une reprise très chaotique en cas de remplacement par une nouvelle personne. L'usage de plus en plus répandu de scénarisations électroniques (à travers une plate-forme de *E-Learning* telle que Moodle², Claroline³, ...) motivé par des besoins d'échange et de réutilisation de ressources pédagogiques, accompagné du besoin d'informatisation des processus de reconfiguration font que ces travaux relèvent des « *e*-TPs », où le *e* est le même que pour le terme *E-Learning* et traduit l'aide apportée par les outils informatiques [1].

L'extension de l'usage de tels systèmes, utilisés actuellement principalement en présentiel, en direction d'un public distant (université partenaire situé à quelques centaines de kilomètres, par exemple, ou même université située dans un pays étranger) est une clef pour se rapprocher de l'objectif de rationalisation de tels systèmes. En effet, cela permet de toucher un public qui ne peut se

1 Voir <http://www.aip-primeca.net/>

2 Voir <http://moodle.org/>

3 Voir <http://www.claroline.net/>

permettre de se déplacer jusqu'au système (pour des contraintes principales de temps) et qui ne peut s'offrir un tel équipement industriel. Dans un tel cas, l'usage des outils du *E-Learning* et plus précisément des télé-TP est indispensable pour assurer la médiation entre le système et les apprenant, et même éventuellement avec le formateur qui peut être également délocalisé.

L'équipe du LIESP a déjà une expérience en terme de téléTP, forgée depuis 2002. Un travail sur l'aide à la réutilisation de scénarios pédagogiques de téléTPs a été abordé en premier lieu au cours de la thèse de doctorat de Hcene BENMOHAMED [2]. Aujourd'hui, Saher ARNOUS, dans le cadre de sa thèse, étudie l'outil de reconfiguration [3] présenté dans cet article. Son objectif est d'intégrer cet outil dans une plate-forme standard de *E-Learning* afin que les enseignants et les étudiants continuent d'utiliser les outils auxquels ils sont habitués. Ces outils, dont beaucoup sont libres et réalisés par une communauté internationale robuste, ont désormais acquis une qualité de service propre à les utiliser en production dans les universités. Au centre de cette batterie est le *Learning Management System* (LMS) : Moodle, Claroline, Ganesha⁴, ... C'est cet outil qui est utilisé au moment même de l'acte de formation, à la fois par les tuteurs et les apprenants. C'est lui qui délivre le contenu pédagogique sous forme électronique et qui est responsable de tracer et évaluer les apprentissages en cours. Il est incontournable.

En amont de cet outil, l'auteur de scénarios pédagogiques (en général un enseignant mais pas forcément le même que celui qui prend le rôle d'instructeur), fait appel à un « outil auteur ». Cet outil, souvent graphique (comme celui intégré à LAMS⁵) et plus (Opale⁶, par exemple) ou moins (*Reload Editor*⁷) convivial pour des non spécialistes du *E-Learning*, aide à l'organisation du scénario, de ses ressources électroniques (liens vers des sites externes, documents attachés (vidéos, PDFs, ...), exercices, ...) et empaquette l'ensemble dans un paquetage assemblé selon un standard (typiquement IMS-CP⁸ pour le format du paquetage et SCORM⁹ pour le format de scénarisation) et compatible avec le LMS qui se chargera de le dépaqueter et d'en délivrer le contenu aux apprenants et tuteurs.

Ce configurateur est donc en étroite collaboration avec l'outil auteur et doit pouvoir transmettre, en plus des programmes automate en directions des Automates Programmables Industriels composant typiquement les systèmes automatisés industriels, les paquetages pédagogiques correspondant à la configuration installée sur le système ciblé. C'est pourquoi une expertise du fonctionnement de ces outils et des standards d'échange d'informations est indispensable dans ce projet.

Fondements liés à la conception de Systèmes Automatisés de Production

Les constructeurs de machines (OEM : *Original Equipment Manufacturer*) sont essentiellement des PME dont les compétences sont plutôt mécaniciennes et centrées sur la commande d'axe, l'objectif visé étant la performance machine. La composante mécanique de leur bureau d'étude a depuis longtemps adopté la conception modulaire en intégrant tout d'abord les outils CAO puis évolué vers le PLM (*Product Lifecycle Management*) afin de satisfaire les exigences de flexibilité et d'évolutivité [4]. La composante automation du bureau d'étude, longtemps restée réservée face aux révolutions informatiques liées à la programmation objet (POO), évolue aujourd'hui et, poussée par les contraintes de coût et temps de conception et d'installation, adopte progressivement des méthodologies combinant structuration et réutilisabilité des programmes automates tout en favorisant la conception collaborative et l'interpénétration des différents métiers associés à la conception.

Ainsi la conception et la réalisation de programmes automates pour SAP, est aujourd'hui de plus

4 Voir <http://www.ganesha.fr>

5 Voir <http://www.lamsfoundation.org>

6 Voir <http://scenari-platform.org/projects/opale/fr/pres/co/>

7 Voir <http://www.reload.ac.uk>

8 Voir <http://www.imsglobal.org/content/packaging>

9 Voir <http://www.adlnet.gov/Technologies/scorm/>

en plus orientée composant [5] (objet sans mécanisme d'héritage, au sens de la POO). Ces composants logiciels codent typiquement des comportements bas-niveau reproduits dans de nombreuses machines (gestion d'un vérin pneumatique ou d'un asservissement de position d'un axe, avec détection d'anomalies, par exemple), comme des comportements plus évolués (synchronisation entre commandes d'axe réalisées chacune par un composant logiciel dédié). Ils sont définis en terme d'entrées/sorties et de comportement une seule fois, puis instanciés autant de fois que nécessaire dans le projet. S'ils ont été conçu de manière assez générique et validés, ils sont réutilisables tels quels dans d'autres projets, ce qui induit un gain de temps de conception, de réalisation et de validation à long terme.

Le configurateur étudié doit donc pouvoir s'inscrire dans cette logique de conception et de réalisation orientée composant pour s'intégrer naturellement dans ces phases d'ingénierie. Nous avons donc étudié en détail ces aspects et notamment la façon de spécifier et représenter des composants à l'aide de SysML [6]. Fortement inspiré de la version 2 d'UML et adopté par l'OMG (*Object Management Group*), SysML est dédié à la problématique de la conception des systèmes complexes. SysML propose neuf types de diagrammes couvrant l'analyse des besoins aux cas de validation (*test case*) en passant par les architectures matérielle et logicielle. Afin de s'appropriier cet outil de représentation, les diagrammes de conception du configurateur ont été réalisés en SysML. L'unification entre la synthèse de programmes automates, de scénarios pédagogiques, de documentations, ... nous a amené à fonder ce projet sur les travaux de thèse de doctorat de Fabien CHIRON [7] proposant une approche Composant pour la synthèse de commande de systèmes automatisés tels que les convoyeurs. Il a proposé une méthodologie basée sur les architectures dirigées par les modèles (*Model Driven Architectures* : MDA) aidant à la génération de code source de programmes automates au standard IEC 61131-3, où chaque composant (au sens *Component Based Automation*, CBA [8]) est décrit par un méta-modèle.

3 Définition des besoins

Besoins généraux

L'objectif principal du configurateur est d'aider le formateur à préparer un SAP et son environnement pédagogique (scénarios, polycopiés, présentations à l'écran, ...) à une activité pratique.

Par contre, il n'est pas dans son rôle d'aider à réaliser cette activité. Tout au plus, il pourra être utilisé pour des reconfigurations du système durant l'activité si cela s'avère nécessaire du point de vue pédagogique mais ceci pourrait être plutôt considéré comme un changement d'activité pratique.

Les acteurs

Les acteurs de ce système sont :

- les acteurs humains :
 - le formateur qui choisit et implante une configuration selon ses objectifs pédagogiques ;
 - l'auteur de configurations qui prépare, en amont, des configurations prêtes à l'emploi ; ce sont des spécialistes dans leur domaine (ordonnancement, robotique, vision, MES, ...) mais pas nécessairement des spécialistes en automation ;



Figure 1. Principaux besoins pour effectuer un e-TP sur une Système Automatisé de Production

- les acteurs non humains :
 - le Système Automatisé de Production (SAP) et plus précisément son ou ses API et Interfaces Homme-Machine industrielles (panels graphiques tactiles par exemple) qui vont être reprogrammés à chaque reconfiguration ;
 - le LMS sur lequel le configurateur va charger les scénarios pédagogiques liés à la configuration visée, et sélectionnés par le formateur.

Cas d'utilisation principal

Partant d'objectifs pédagogiques (formation sur la traçabilité à partir d'étiquettes RFID, par exemple), le formateur doit pouvoir, à travers ce système :

- choisir une configuration compatible avec ses objectifs pédagogiques à travers un moteur de recherche puisant dans un catalogue de formations ;
- charger les programmes automates correspondant sur le système automatisé ciblé ;
- charger les scénarios pédagogiques liés, sur un LMS (celui de son institution ou éventuellement un LMS spécifique) ;
- télécharger la documentation complémentaire : pour les opérations manuelles de configuration non automatisables et autres notices d'utilisation.

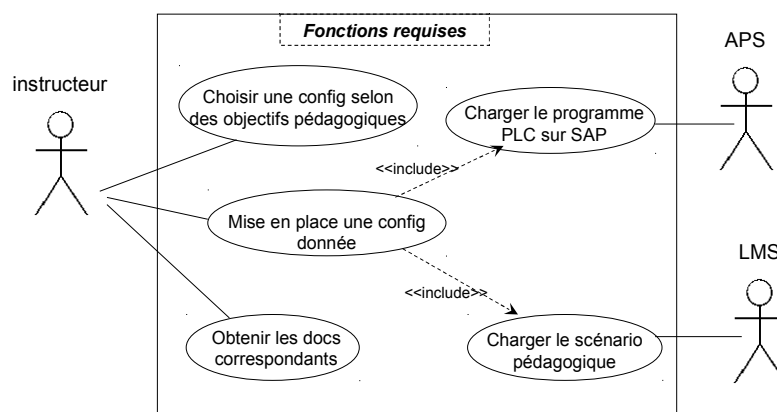


Figure 2. Cas d'utilisation principaux

Éléments de conception générale

Ce chapitre n'a pas pour vocation de présenter exhaustivement toute l'étape de conception générale mais d'apporter quelques détails importants.

Choix du standard de *E-Learning*

Afin de dialoguer avec des LMS, il a été nécessaire d'opter pour un standard de *E-Learning*. Il en existe plusieurs actuellement supportés par les LMS libres et commerciaux :

- *Sharable Content Object Reference Model* (SCORM): [9], utilisé pour créer, échanger sous forme de paquetage, et faire tourner des scénarios pédagogiques ;
- *IMS Global Learning Consortium Learning Design* (IMS-LD) [10], semblable à SCORM mais plus flexible ;
- *IMS Global Learning Consortium Common Cartridge* [11] plus récent et à mi-chemin entre SCORM et IMS-LD.

Nous avons opté pour le standard SCORM car celui-ci est aujourd'hui le plus répandu sur les LMS libres et commerciaux.

Redéfinition des acteurs

L'étude approfondie des besoins a démontré la nécessité d'ajouter des acteurs pour que le système soit utilisable par les utilisateurs finaux qui sont les formateurs sans obligation d'expertise en automatisme. Ainsi, nous avons dû ajouter un nouvel acteur humain : l'expert en automation, chargé de développer et maintenir les programmes API et aider les auteurs à concevoir des configurations. Nous avons spécialisé l'acteur LMS qui doit désormais être compatible SCORM, ce qui n'est pas une contrainte forte. Et, afin de ne pas avoir à reconcevoir tout un système de gestion de maquettes pédagogiques de TP, nous avons ajouté un « ELAMS » (*Elab Management System* : Système de Gestion de Maquettes de e-TP) développé au cours des travaux d'Hcene Benmohamed. Du point de vue du configurateur, l'ELAMS a pour fonction de gérer les communications entre les API d'une part et le LMS et les acteurs humains d'autre part (à travers des interfaces homme machine (IHM) web dédiées).

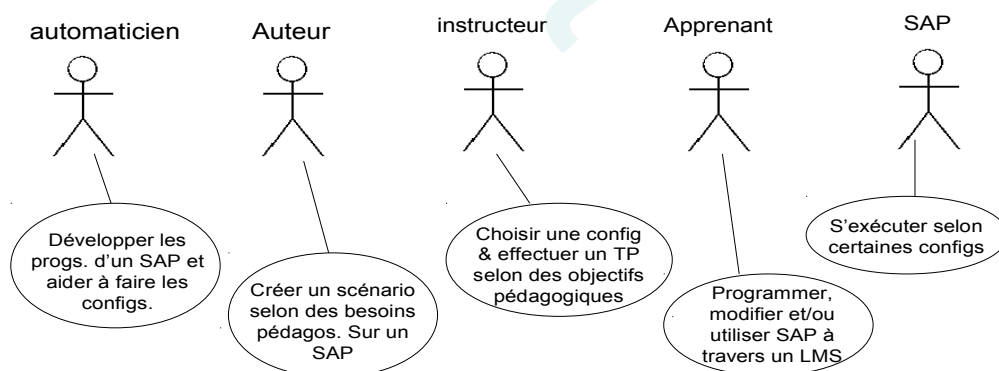


Figure 3. Ensemble finalisé des acteurs du reconfigurateur

Cas d'utilisation complémentaire

Le cas d'utilisation présenté ici complète le précédent et se situe au niveau temporel en amont de l'utilisation du configurateur pour lancer une configuration : il s'agit ici de préparer des configurations pour un usage ultérieur.

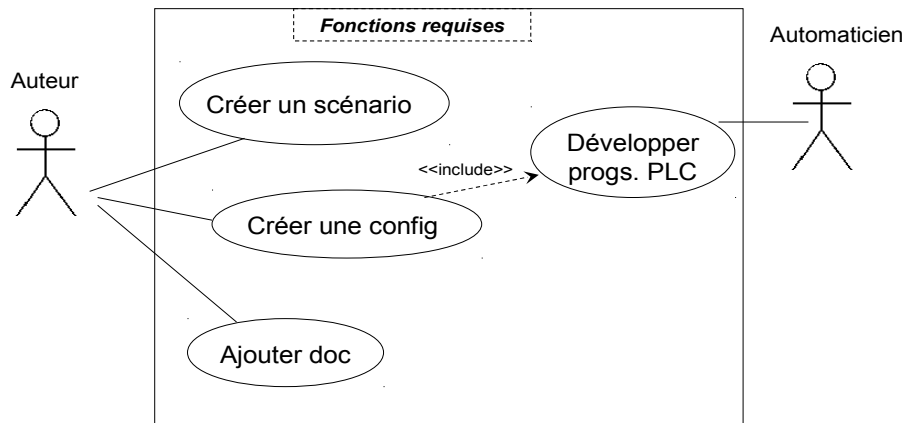


Figure 4. Cas d'utilisation lié à l'expert en automatisme

Ce qui permet de préciser les sous-besoins attendus de ce configurateur :

- un outil auteur permettant de créer de nouvelles configurations liées à des objectifs pédagogiques ;
- une bibliothèque de configurations avec un moteur de recherche basé sur les objectifs pédagogiques ;
- un installateur permettant d'automatiser au maximum l'installation des programmes API, scénario sur LMS et autres données de sortie.



Figure 5. Besoins fonctionnels affinés, recentrés sur le configurateur

Flux de données

Les données d'entrée du système, que se soit pour l'**outil auteur de création de configurations** que pour l'**outil de choix et d'installation de configurations**, est un ensemble de besoins pédagogiques. Ceux-ci sont exprimés en terme de compétences à acquérir, par exemple : « *extraire des données d'un SAP pour calculer son Taux de Rendement Synthétique (TRS) à travers un Manufacturing Execution System (MES)* », « *Mise en place d'un système de traçabilité basée sur des étiquettes RFID* », « *Maîtriser le logiciel UNITY de Schneider Electric* », ...

Les données de sortie sont :

- les programmes API (dont le format est dépendant du modèle et du fournisseur des API)

ciblés) ;

- le packaging SCORM à destination du LMS, qui code la scénarisation liée à cette activité d'apprentissage pratique ;
- la documentation complémentaire explicitée ci-dessus ;
- la configuration ELAMS correspondante, nécessaire pour faire communiquer le SAP et le LMS.

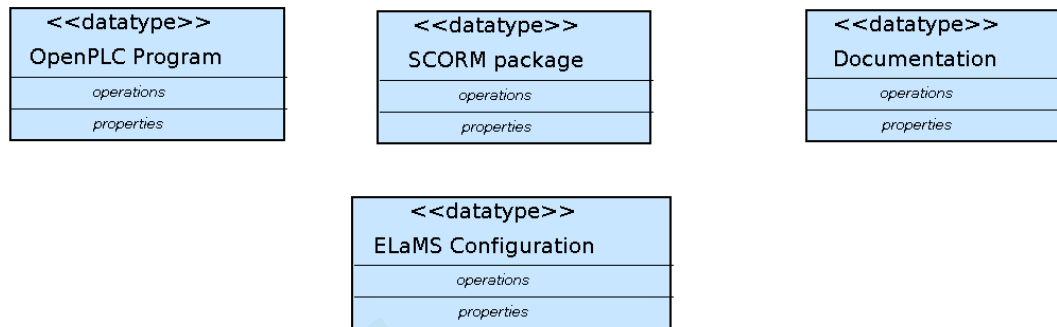


Figure 6. Types principaux de données en entrée et en sortie

Décomposition fonctionnelle

Afin de pouvoir faire appel à une programmation orientée composant et réutiliser ces composants dans plusieurs configurations, il a été nécessaire d'élaborer une décomposition de ces données. Ainsi, l'élément atomique de cette décomposition est appelé « composant » (*component*) et est constitué de l'assemblage de :

- un sous-programme API (une configuration matérielle, une liste de variables globales, un composant réutilisable (appelé *DFB* dans *Unity*), une section de code API, ... stocké au format OpenPLC pour en favoriser la réutilisation ;
- un ensemble de scénarios pédagogiques élémentaires en liaison avec ce sous-programme ;
- une documentation liée à l'explication et l'utilisation de ce composant ;
- un ensemble de définitions d' IHM et ;
- des métadonnées permettant de préciser l'auteur, la version, sa nature, des mots clefs, l'usage, ... des objectifs pédagogiques liés, ...

Ainsi, il est nécessaire de préciser la décomposition fonctionnelle du système :

- un outil auteur de création, d'édition et d'assemblage de composants et
- un outil de génération de configurations finales.

D'autre part, afin de créer des composants, il est nécessaire de faire appel à des éditeurs des bibliothèques pour chaque sous-type de données (programme API, documentation, scénario élémentaire, ...) et pour les composants eux-même.

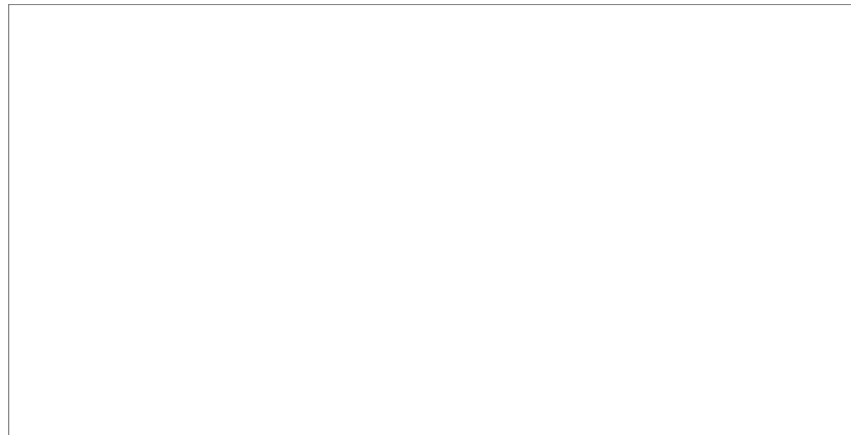


Figure 7. Décomposition des types de données internes

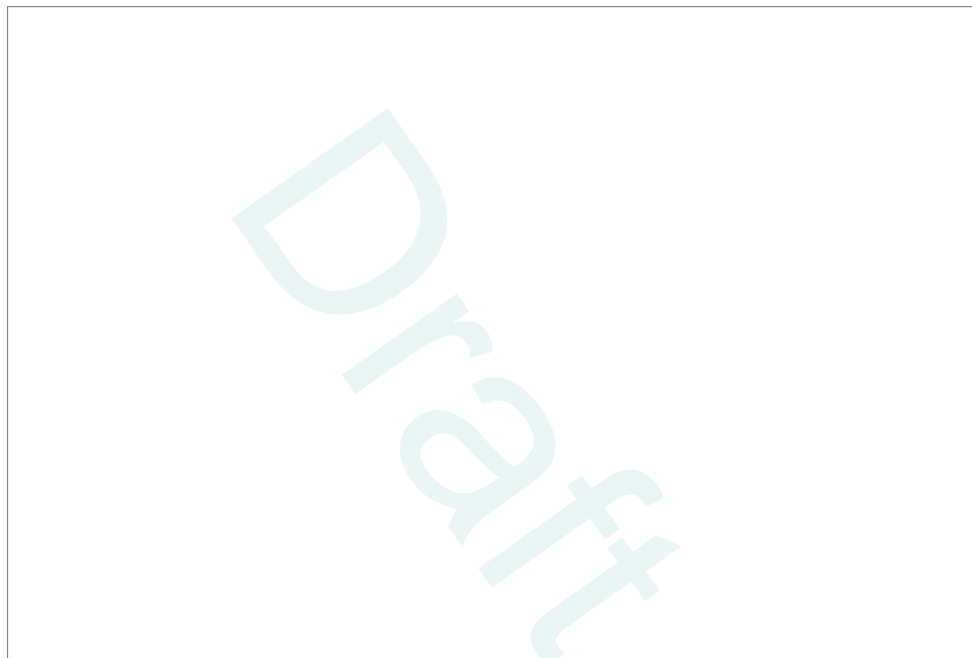


Figure 8. Décomposition fonctionnelle de l'outil auteur

4 Éléments de prototypage

Ce système étant relativement complexe, nous avons tenté de réutiliser des outils existants. Ainsi, pour tout l'aspect ingénierie pédagogique, nous faisons appel à la chaîne éditoriale pédagogique OpaleSup. Sa version client-serveur et le codage sous forme de fichier XML de ses éléments de scénarios pédagogiques en fait un parfait partenaire pour notre configurateur. Ainsi l'auteur se retrouvera dans un environnement familier pour cet aspect de la conception.

Pour l'aspect programmation d'API, tous nos automates étant fournis par Schneider Electric, nous utiliserons leur logiciel de développement de programmes API Unity. L'interfaçage est par contre plus difficile étant donné qu'il s'agit d'un logiciel propriétaire. Ainsi nous utiliserons les fonctions d'export et d'import des données (configurations, DFB, listes de variables, sections, ...) qui génèrent des fichiers XML éditables ensuite par notre outil pour effectuer automatiquement des assemblages et donc des squelettes de programmes. La finition, manuelle, sera alors effectuée par l'expert en automatismes et le fichier final au format propriétaire enregistré pour être téléchargé ultérieurement

sur un API. Certes, l'usage intensif des fonctions d'import/export n'est pas très ergonomique mais il s'agit ici de réaliser un prototype à des fins de validation et non un logiciel utilisable tel quel dans un environnement de production.

5 Expérimentation

Choix du Système Automatisé de Production

L'expérimentation aura lieu sur le Tranfert Libre de l'AIP Primeca RAO¹⁰ localisé à Lyon.



Figure 9. Le transfert libre de l'AIP Primeca RAO utilisé pour les expérimentations

Il s'agit d'un Système Automatisé de Production à visée pédagogique composé d'un système de transitique, de deux robots industriels d'assemblage et d'un système de contrôle visuel par caméra. Il est utilisé comme support pratique pour de nombreux établissements de la région Rhône-Alpes pour des besoins pédagogiques variés. La production de ce système consiste à poser et déposer des pièces de différentes couleurs sur des palettes équipées d'étiquettes électroniques passives.

Les raisons de ce choix sont que :

- il s'agit d'un SAP complexe (donc bien adapté au besoin d'un configurateur) avec six postes semblables (propriété propice à la réutilisation de composants élémentaires) spécialisés dans des fonctions différentes (leur nombre rend une (re)configuration manuelle trop longue pour être lancée en début de séance, voire plusieurs fois par séance) ;
- il a été tout récemment migré de PL7Pro vers Unity avec une reprogrammation complète basée sur des composants logiciels (pas besoin d'une nouvelle réécriture des programmes pour cette expérimentation) ;
- il est utilisé pour des besoins pédagogiques variés : supervision, automatisme, ordonnancement, ... et par des enseignants issus d'instituts différents (une petite dizaine) ;
- à certaines périodes de l'année, il est très utilisé, ce qui oblige parfois d'intervenir hors horaires scolaires pour préparer une session ;
- certains enseignants sont autonomes mais nombreux sont ceux qui se reposent sur l'ingénieur responsable de la plate-forme pour configurer le système. Un logiciel leur permettant de charger les configurations limiterait les risques de blocage en l'absence du responsable ;
- cette même personne est la seule à connaître et maîtriser l'ensemble des configurations des

10 Voir <http://aiprao.insa-lyon.fr>

différents usagers ; ce qui l'immobilise régulièrement pour des tâches répétitives et induit un risque important de ne pas pouvoir utiliser le système en son absence ;

- les ressources documentaires et différentes versions de programmes ont été mutualisés au moment de la migration mais, avec le temps, chaque enseignant va probablement faire évoluer ses programmes, ses sujets de TP, ... sans en faire profiter l'ensemble de la communauté, ce qui induit une perte de temps pour chacun (le configurateur avec son outil auteur permettrait ainsi de partager et réutiliser l'ensemble des ressources pédagogiques, pas uniquement les programmes).

Déroulement prévu

L'objectif de cette expérimentation est de prouver que le configurateur est techniquement viable d'une part (il remplit les fonctions que l'on attend de lui) et apporte une plus-value par rapport à l'existant, d'autre part. Nous allons donc prendre contact avec des enseignants de disciplines différentes utilisateurs du transfert libre :

- en automation : où il s'agit de programmer tout ou partie de ce système ;
- en ordonnancement: pour une utilisation de ce système ;
- MES : où il s'agit d'adapter ce système pour qu'il puisse échanger des données avec un MES.

Nous allons les faire accompagner par un expert en automation et leur demander de concevoir une configuration répondant à leurs besoins et de tester la phase de choix et d'installation de leur configuration. Une évaluation leur sera demandée du point de vue de l'ergonomie du configurateur.

6 Conclusion et perspectives

Cet article a présenté l'avancement d'un travail de thèse démarré fin 2008 au laboratoire LIESP. Il s'agit d'un configurateur de SAP à des fins pédagogiques. Nous avons tenu compte d'un double usage : par des industriels soucieux d'améliorer leurs processus de préparation à la formation de personnels à l'usage d'un SAP, ainsi que par des enseignants utilisant un SAP partagé entre plusieurs utilisations. Afin de rester au plus près des méthodologies industrielles, nous avons adopté le standard SysML pour représenter le modèle de ce configurateur lors de sa conception. Nous sommes actuellement en phase de réalisation d'un prototype dont l'objectif est de démontrer la faisabilité technique d'un tel logiciel. Son usage lors d'expérimentations nous aidera à démontrer sa plus-value dans le contexte générique de l'AIP Primeca.

Ces travaux offrent de nombreuses voies d'amélioration et d'extension. En premier lieu, il serait intéressant de s'appuyer sur un environnement de programmation d'automates plus générique, tel que Control Build de la société Geensys, capable de traiter le développement de programmes API indépendamment du modèle et du fournisseur de l'API ciblé. Par ailleurs, ce système étant prévu pour être utilisé à travers un LMS, il est tout à fait envisageable d'étendre son usage local à un usage à distance [12] et ainsi en améliorer son taux d'usage et sa rentabilité ; cela nécessiterait de compléter les composants actuels par des éléments d'IHM permettant de piloter le SAP depuis un navigateur web sans obligation de présence locale.

Références

- [1] M. ASHISH et C. PATVARDHAN, "A study of ICT enabled laboratories", Actes de la conférence annuelle IEEE INDIA, page 1, ISBN 1-4244-0369-3, doi: 10.1109/INDCON.2006.302765, New Delhi, Inde, 15-17 sept. 2006.
- [2] H. BENMOHAMED, "ICTT@Lab : un environnement pour informatique la génération et l'exécution de scénarios de téléTPs", Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, 2007.
- [3] S. ARNOUS, A. LELEVE, K. KOUISS, P. PREVOT, "Towards semi-automatic generation

- of training scenarios in industrial automated systems", Actes de l'*International Conference on Management of Emergent Digital EcoSystems*, ACM New York, NY, USA, doi:10.1145/1643823.1643914, pp. 74, Lyon, 27-30 octobre 2009
- [4] ARC Advisory Group, "Valorisation des machines par la programmation modulaire", revue *J'automatise*, groupe Cimax, n°48, 2006, pp. 6-7.
- [5] K. TRKAJ, "Users introduce component based automation solutions", *Computing and Control Engineering Journal*, Vol. 15, N° 6, ISSN: 0956-3385, pp. 32-37, janvier 2005.
- [6] P. ROQUES, "SysML par l'exemple, un langage de modélisation pour systèmes complexes", Eyrolles, ISBN 9782212850062, 2009.
- [7] F. CHIRON, "Contribution à la flexibilité et à la rapidité de conception des systèmes automatisés avec l'utilisation d'UML", Thèse de doctorat, Université Blaise Pascal de Clermont- Ferrand, 2007.
- [8] J.L. LALLICAN , P. BERRUET, A. ROSSI et J.L. PHILIPPE, "A component-based approach for conveying systems control design", Actes de la conférence ICINCO-SPSMC, INSTICC Press, ISBN 978-972-8865-84-9, pp. 329-338, Angers, France, 9-12 mai 2007.
- [9] Advanced Distributed Learning, "Sharable Content Object Reference Model (SCORM)", <http://www.adlnet.gov>, 2004
- [10] IMS Global Learning Consortium, "IMS Learning Design Specification", <http://www.imspjproject.org/learningdesign/>, 2003
- [11] IMS Global Learning Consortium, "IMS Common Cartridge Specification", <http://www.imsglobal.org/commoncartridge.html>, 2006
- [12] O.G. BELLMUNT, D.M. MIRACLE, S.G. ARELLANO, A. SUMPER et A.S. ANDREU, "A distance plc programming course employing a remote laboratory based on a flexible manufacturing cell", revue *IEEE Transactions on Education*, vol. 49, no. 2, pp. 278-284, 2006